



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,770	10/24/2003	Christian Zander	7469 US	8945
71012 Fogarty, L.L.C. P.O. Box 703695 Dallas, TX 75370-3695	7590 07/07/2009			
EXAMINER				
FEARER, MARK D				
ART UNIT		PAPER NUMBER		
2443				
MAIL DATE		DELIVERY MODE		
07/07/2009		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/692,770

Applicant(s)

ZANDER, CHRISTIAN

Examiner

MARK D. FEARER

Art Unit

2443

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 23 March 2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 17-36 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 17-36 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SE/US)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. Applicant's Amendment filed 23 March 2009 is acknowledged.
2. Claim 17 is amended.
3. Claims 17-36 are pending in the present application.
4. This action is made FINAL.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

6. Claims 17-19, 22, 25-26, 28-31, and 34-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nguyen et al. (US 6304915 B1) in view of Ehrhardt et al. (US 20010015732 A1).

Consider claim 17. Nguyen et al. discloses a method of setting up a communication procedure that is executable on a protocol tester using message sequence charts (MSC) to graphically display the communication procedure between two instances, comprising: creating a configuration file to be used by the protocol tester, the configuration file including one or more new functions that are not available in an original version of the protocol tester provided by a manufacturer, the configuration file comprising for each of the new functions ((“In the directory structure defined below, documents are stored corresponding to the preferences. The top level of the directory structure is the content-type, the next level is language (for NLS support). For example, to create text/html content in US English & French, the directory structure given below would contain the HTML documents for each of the transactions. The VPOS terminal cartridge has a configuration file that allows the user to specify the content-type as well as the language to be used for a cartridge. The first release of the VPOS terminal cartridge supports one content-type and language for each server.”) column 35 lines 19-29 (“When the object or class representing the ceramic piston engine inherits all of the aspects of the objects representing the piston engine, it inherits the thermal characteristics of a standard piston defined in the piston engine class. However, the ceramic piston engine object overrides these ceramic specific thermal characteristics,

which are typically different from those associated with a metal piston. It skips over the original and uses new functions related to ceramic pistons. Different kinds of piston engines have different characteristics, but may have the same underlying functions associated with it (e.g., how many pistons in the engine, ignition sequences, lubrication, etc.). To access each of these functions in any piston engine object, a programmer would call the same functions with the same names, but each type of piston engine may have different/overriding implementations of functions behind the same name. This ability to hide different implementations of a function behind the same name is called polymorphism and it greatly simplifies communication among objects.”) column 8 lines 28-46): a name for the new function (“FIG. 18C is an example flowchart of message processing in accordance with a preferred embodiment. Processing commences at function block 1880 when a message is received by an HTTPS server or other listener and passed to decision block 1883 to determine if the sending VPOS has transmitted an authentic message and if the VPOS is authorized to communicate with this gateway. If the message is not authentic, then the message is logged as an error and the error is handled as shown in function block 1889. If the message is authentic, then the message is decrypted at function block 1884 and the PDU parses the message into name/value pairs. Then, based on the message type and the extended SET version information, the remaining message is parsed at function block 1885 and the message is checked for conformance to the appropriate specification as shown at decision block 1887. If the message does not conform, then it is logged and the error handled at function block 1889. If the message conforms to the proper specification in decision block 1887 then

the message is translated into the appropriate host format and sent to the host as shown in function block 1888. Thus, when a gateway receives an incoming message from a VPOS and parses the Extended SET portion of the message, a single MIME message can transmit a SET message and/or an Extended Set Message.") column 57 line 53 – column 58 line 9); a data type for the new function ("Banks desire an Internet payment solution that emulates existing Point of Sale (POS) applications that are currently installed on their host computers, and require minimal changes to their host systems. This is a critical requirement since any downtime for a bank's host computer system represents an enormous expense. Currently, VeriFone supports over fourteen hundred different payment-related applications. The large number of applications is necessary to accommodate a wide variety of host message formats, diverse methods for communicating to a variety of hosts with different dial-up and direct-connect schemes, and different certification around the world. In addition, there are a wide variety of business processes that dictate how a Point of Sale (POS) terminal queries a user for data and subsequently displays the data. Also, various vertical market segments, such as hotels, car rental agencies, restaurants, retail sales, mail sales/telephone sales require interfaces for different types of data to be entered, and provide different discount rates to merchants for complying with various data types. Moreover, a plethora of report generation mechanisms and formats are utilized by merchants that banking organizations work with.") column 3 line 50 – column 4 line 4); an identification of a new description file, the description file comprising code to be used to provide the new function ("A Data Manager provides storage and retrieval of generic

data items and database records. It is assumed that data fields, index fields or entire data records can be marked as encrypted and the encryption process is largely automated. The data manager has no specific knowledge of database records appropriate to different payment methods. This layer is separated out so as to reduce changes required when new payment methods are introduced. However RSA key pairs and certificates might be considered as "simple" data types. This component also provides an abstraction which supports wallet files on computer disk or contained in smart cards.") column 83 lines 38-48 ("A preferred embodiment includes a single file or directory of files comprising a "wallet" which contains personal information and information about multiple payment methods with the preferred implementation. These payment methods (Visa cards, debit cards, smart cards, micro-payments etc.) also contain information such as account numbers, certificates, key pairs, expiration dates etc. The wallet is envisaged to also contain all the receipts and transaction records pertaining to every payment made using the wallet. A Cryptographic API component provides a standard interface for RSA and related cryptographic software or hardware. This support includes encryption, signature, and key generation. Choice of key exchange algorithm, symmetric encryption algorithm, and signature algorithm should all be configurable. A base class stipulates generic behavior, derived classes handle various semantic options (e.g. software based cryptography versus hardware based cryptography.)") column 83 line 55 – column 84 line 5); and an identification of a new graphic symbol file for the new function, the graphic symbol file corresponding to a new graphic symbol to be displayed on the protocol tester to allow selection of the new

function when setting up the communication procedure ((“A brief description of the Virtual Point of Sale Terminal cartridge functions are provided below. VPOSTInit(), VPOSTExec() and VPOSTShut() are the entry points required for each cartridge in accordance with a preferred embodiment. The other functions implement some of the key VPOST cartridge functionality. A source listing of the VPOS code is provided below to further accentuate the detailed disclosure of a preferred embodiment.”) column 35 lines 32-40 (“In the block diagram shown in FIG. 15B, the VPOS provides an interface for transactions which are initiated both by the consumer and the merchant. The merchant initiates a transaction from a Graphical User Interface (GUI) 1550 and all the transactions that are initiated by the consumer are routed by the Merchant WEB Server 1545. The Authorization/Data Capture Module 1560 processes the requests originated by the merchant or the consumer and routes them to the Protocol Module 1565. The Protocol Module is responsible for building the payment protocol request packet (e.g., an SSL-encapsulated ISO 8583 packet) 1570 before sending the request to the Gateway 1579. Then, the Gateway 1579 awaits a response from the Protocol Module 1565, and upon receiving the response, the Gateway 1579 parses the data and provides unwrapped data to the Authorization/Data-Capture Module 1560. The Authorization/Data-Capture Module 1560 analyzes the response and updates the Transaction Log 1580. The Transaction Log 1580 contains information concerning any successfully completed transactions and the accumulators or the transaction totals. The VPOS terminal creates and maintains the Transaction Log 1580, and the VPOS Configuration Data 1585 contains information which is used to configure the behavior of

the VPOS. The entire VPOS functionality is thread-safe and hence using the VPOS in a multi-threaded environment does not require any additional interfacing requirements.”) column 36 line 66 – column 37 line 25); selecting instances involved in the communication procedure ((“The merchant, interacting with the extended terminal admin page, navigates to the list of gateways which is maintained by the Test Gateway, and selects the bank to connect by selecting from the list of banks, at function block 5018. During this process, the merchant's public key certificates are uploaded to the Test Gateway, and checked (at decision block 5025) to verify that the certificates have been signed by the bank to customize the bank for the VPOS. If the certificates do not match, the merchant is advised of the situation in function block 5028, and must select a different bank. If the certificates are not valid SET certificates as detected at decision block 5020, the merchant is advised at function block 5028, and the session terminates. If the certificates are valid and match the selected bank, customization continues at function block 8030.”) column 60 lines 51-65); selecting a protocol layer based upon the selected communication procedure ((“FIG. 15A illustrates a payment processing flow in accordance with a preferred embodiment. The payment functionality provided by the VPOS terminal is divided into two main categories: "Merchant-Initiated" 1510 and "Consumer-Initiated" 1500. Some payment transactions require communication with the acquirer bank through the Gateway 1530. The normal flow of a transaction is via the VPOS Cartridge API 1512 to the VPOS C++ API 1514 into the payment protocol layer 1516 which is responsible for converting into the appropriate format for transmission to the Gateway for additional processing and forwarding to existing host payment

authorization systems. Host legacy format refers to an existing authorization system for credit card approval currently utilized with the VeriFone Point of Sale (POS) gray terminals. The output from the payment protocol layer 1516 is transmitted to the authorization processing center via the gateway 1530. These transactions are referred to as "Online Transactions" or "Host Payments." The transactions that can be done locally by the merchant without having to communicate with the acquirer bank are referred to as "Local Functions and Transactions." To support different types of payment instruments, the VPOS Terminal payment functionality is categorized as set forth below.") column 24 lines 22-45); selecting abstract communication interfaces of the protocol layer for the communication procedure ("FIG. 1B depicts an overview of the present invention. Customer computer system 120 is in communication with merchant computer system 130. The customer-merchant session 150 operates under a general-purpose secure communication protocol such as the SSL protocol. Merchant computer system 130 is additionally in communication with payment gateway computer system 140. A payment gateway is a system that provides electronic commerce services in support of a bank or other financial institution, and that interfaces to the financial institution to support the authorization and capture of transactions. The customer-institution session 170 operates under a variant of a secure payment technology such as the SET protocol, as described herein, referred to as Merchant-Originated Secure Electronic Transactions ("MOSET"), as is more fully described herein.") column 13 lines 18-32); setting up the communication procedure through the protocol tester based on the selections made in the selecting steps, with abstract communications interfaces

selecting and the communication data selecting steps begin made graphically; selecting the new function while setting up the communication procedure by graphically selecting the new graphic symbol ("When the object or class representing the ceramic piston engine inherits all of the aspects of the objects representing the piston engine, it inherits the thermal characteristics of a standard piston defined in the piston engine class. However, the ceramic piston engine object overrides these ceramic specific thermal characteristics, which are typically different from those associated with a metal piston. It skips over the original and uses new functions related to ceramic pistons. Different kinds of piston engines have different characteristics, but may have the same underlying functions associated with it (e.g., how many pistons in the engine, ignition sequences, lubrication, etc.). To access each of these functions in any piston engine object, a programmer would call the same functions with the same names, but each type of piston engine may have different/overriding implementations of functions behind the same name. This ability to hide different implementations of a function behind the same name is called polymorphism and it greatly simplifies communication among objects.") column 8 lines 27-46 ("Objects can represent elements of the computer-user environment such as windows, menus or graphics objects.") column 8 lines 58-60); and associating parameters selectable for the new function with the new description file ("The interaction between the VPOS and a client commences when a pay page solicits parameters of a transaction. Then, the parameters are validated to be sure the payment instrument, for example, cardnumber is not null. Then, a transaction object is created, eg. AUTHONLY, and the object is initialized and stuffed with parameters of the

transaction, eg. `ao.setpan(accnum)`, and the object is executed. This execution invokes the VPOS engine. The VPOS engine further validates the parameters based on the particular merchant's configuration. For example, some merchants do not accept American Express Cards, but will take Visa, and all merchants check the expiration date of the card. Assuming a valid and acceptable card has been tendered, then a TID is assigned (expiring, existing TIDs) or block a new TID from the TID Pool. This generates a STAN, XID, RRPID unique tag and creates an initial record in the transaction database which is flagged as before gateway processing in case the transaction crashes and must be backed out. Then the protocol parameters are identified in the registry based on card type, and a particular acquirer identified. Then, a protocol object is created and executed to extract results from the protocol object and the before gateway "bit" is flipped to again flag the location of the transaction in the process as it is submitted to the Gateway.") column 63 lines 10-36).

However, Nguyen et al. fails to disclose a system or method comprising a method of graphically modifying a display interface on a protocol tester.

Ehrhardt et al. discloses a system and method of setting up a communication procedure between instances and a protocol tester comprising a method of graphically modifying a display interface on a protocol tester ("FIG. 1 shows a graphical user interface (GUI) 10 that allows in a first step graphically selecting instances taking part in a communication procedure. Graphical selection in connection means that a symbol or a text proposal is shown graphically on the GUI, such as on a personal computer (PC) screen, and may be selected by simple activation, i.e., by clicking on it with a "mouse."

One of the instances is a protocol tester on which the method as described herein is made available, with the protocol tester in the present case emulating a component, TC.sub.--1. Using two buttons, "Add" 12 and "Delete" 14, a user may add further instances or delete instances listed. In a field 16 the compilation of instances is listed, while in another field 18 the compilation is shown as a diagram. In another field 19 the name of the instance may be selected, and in a further field 20 the instance type is shown. Two buttons, "Back" 22 and "Next" 24, allow the user to move from one level of the definition of the communication procedure to the next, both in the direction of more detailed specifications and in the direction of higher-level presentations. A "Cancel" button 26 allows leaving a level, meaning that the changes made are reset. A "Help" button 28 offers the user further support.") Ehrhardt et al., paragraph 0014).

Nguyen et al. discloses a prior art method of setting up a communication procedure that is executable on a protocol tester using message sequence charts (MSC) to graphically display the communication procedure between two instances, comprising: creating a configuration file to be used by the protocol tester, the configuration file including one or more new functions that are not available in an original version of the protocol tester provided by a manufacturer, the configuration file comprising for each of the new functions; a name for the new function; a data type for the new function; an identification of a new description file, the description file comprising code to be used to provide the new function; and an identification of a new graphic symbol file for the new function, the graphic symbol file corresponding to a new graphic symbol to be displayed on the protocol tester to allow selection of the new

function when setting up the communication procedure; selecting instances involved in the communication procedure; selecting a protocol layer based upon the selected communication procedure; selecting abstract communication interfaces of the protocol layer for the communication procedure; setting up the communication procedure through the protocol tester based on the selections made in the selecting steps, with abstract communications interfaces selecting and the communication data selecting steps begin made graphically; selecting the new function while setting up the communication procedure by graphically selecting the new graphic symbol; and associating parameters selectable for the new function with the new description file upon which the claimed invention can be seen as an improvement.

Ehrhardt et al. teaches a prior art comparable system and method of setting up a communication procedure between instances and a protocol tester comprising a method of graphically modifying a display interface on a protocol tester.

Thus, the manner of enhancing a particular device (system and method of setting up a communication procedure between instances and a protocol tester comprising a method of graphically modifying a display interface on a protocol tester) was made part of the ordinary capabilities of one skilled in the art based upon the teaching of such improvement in Ehrhardt et al. Accordingly, one of ordinary skill in the art would have been capable of applying this known improvement technique in the same manner to the prior art method of setting up a communication procedure that is executable on a protocol tester using message sequence charts (MSC) to graphically display the communication procedure between two instances, comprising: creating a configuration

file to be used by the protocol tester, the configuration file including one or more new functions that are not available in an original version of the protocol tester provided by a manufacturer, the configuration file comprising for each of the new functions; a name for the new function; a data type for the new function; an identification of a new description file, the description file comprising code to be used to provide the new function; and an identification of a new graphic symbol file for the new function, the graphic symbol file corresponding to a new graphic symbol to be displayed on the protocol tester to allow selection of the new function when setting up the communication procedure; selecting instances involved in the communication procedure; selecting a protocol layer based upon the selected communication procedure; selecting abstract communication interfaces of the protocol layer for the communication procedure; setting up the communication procedure through the protocol tester based on the selections made in the selecting steps, with abstract communications interfaces selecting and the communication data selecting steps begin made graphically; selecting the new function while setting up the communication procedure by graphically selecting the new graphic symbol; and associating parameters selectable for the new function with the new description file of Nguyen et al. and the results would have been predictable to one of ordinary skill in the art, namely, one skilled in the art would have readily recognized a system and method of selecting a protocol layer for a communication session.

Consider claim 18, as applied to claim 17. Nguyen et al., as modified by Ehrhardt et al., discloses a method wherein for each parameter of the new function and for a

result of the new function, entering into the configuration file a name and a data type (Nguyen et al., column 29 line 64 – column 30 line 2).

Consider claim 19, as applied to claim 17. Nguyen et al., as modified by Ehrhardt et al., discloses a method wherein the graphic symbol is selected from the group consisting of: a display name of the new function, and an icon for the new function, the graphic symbol file comprising display data for the graphic symbol ((FIG. 61 depicts a flow diagram for the Gwdb_UpdateSetMsgResponseInfo routine. Execution begins at Step 6100. In Step 6110, the routine invokes a database update function by, for example, executing an SQL UPDATE command. In Step 6120, the database return code is obtained in order to be used as a return code from the Gwbd_UpdateSetMsgResponseInfo routine. In Step 6190, the routine returns control to the calling program.") Nguyen et al., column 99 line 66 – column 100 line 6 ("FIG. 62 is the main administration display for the Gateway in accordance with a preferred embodiment. A set of menu selections are presented at 6200 which will be described in more detail for each display. FIG. 63 is a configuration panel in accordance with a preferred embodiment. The configuration panel provides access to management information for configuring a gateway management information database. The Merchant Identifier (Mid) 6310 is a thirty character, alphanumeric field that uniquely defines a merchant. The Merchant Name 6320 is a fifty character, alphanumeric field, the Edit 6330 and Delete field 6340 are hyperlinks to detailed panels for modifying information in the management information database. FIG. 64 is a host communication display for

facilitating communication between the gateway and the acquirer payment host. The IP Address Field 6410 contains the Internet Protocol address for communicating via TCP/IP to the Internet. The TCP logical port field 6430 uniquely identifies the port for accessing the Internet, and the SAVE field 6430 invokes storing of the host communication information in the database. FIG. 65 is a Services display in accordance with a preferred embodiment. This display initiates portions of the Gateway such as the host multiplexer 2130 of FIG. 21. FIG. 66 is a graphical representation of the gateway transaction database in accordance with a preferred embodiment. Each of the fields represents a portion of the internet database schema in accordance with a preferred embodiment.") Nguyen et al., column 100 lines 38-64).

Consider claim 22, as applied to claim 17. Nguyen et al., as modified by Ehrhardt et al., discloses a method wherein data for the new functionality is entered in the configuration file (Nguyen et al., column 29 line 64 – column 30 line 2).

Consider claim 25, as applied to claim 17. Nguyen et al., as modified by Ehrhardt et al., discloses a method wherein the instances involved in the communication are graphically selected and the protocol layer is graphically selected ("The payment manager 2730 also sends and receives transactions to the protocol manager 2770 including a merchant's payment message 2760, a consumer certificate and PK handle 2750, a merchant URL 2742, a payment 2740, a signed receipt 2734 and a GSO, Selected Payment Protocol and Selected Payment Instrument 2732. The payment

manager 2730 also accepts input from the payment applet or MIME message from the merchant as shown at function block 2780. One aspect of the payment processing is a Consumer Payments Class Library (CPCL) 2770 which encapsulates the payment protocols into a single API. By encapsulating the payment protocols, applications are insulated from protocol variations. A SET Protocol provides an implementation of the client-side component of the Secure Electronic Transaction (SEM) Protocol. A complete implementation of the client-side component of the CyberCash micro-payment protocol is also provided.") Nguyen et al., column 82 line 66 – column 83 line 15).

Consider claim 26, as applied to claim 17. Nguyen et al., as modified by Ehrhardt et al., discloses a method wherein the abstract communication interfaces comprise Service Access Points (SAP) (Nguyen et al., column 62 line 15-40).

Consider claim 28, as applied to claim 17. Nguyen et al., as modified by Ehrhardt et al., discloses a method wherein the communication data selecting step comprises: graphically selecting a data format; and graphically setting up a communication sequence between the instances involved (Nguyen et al., column 82 line 66 – column 83 line 15).

Consider claim 29. Nguyen et al., as modified by Ehrhardt et al., discloses a protocol tester for testing a communication procedure, comprising: means for creating a configuration file to be used by the protocol tester, the configuration file including one or

more new functions that are not available in an original version of the protocol tester provided by a manufacturer, the configuration file comprising for each of the new functions (Nguyen et al., column 35 lines 19-29, and column 8 lines 28-46): a name for the new function (column 57 line 53 – column 58 line 9); a data type for the new function (Nguyen et al., column 3 line 50 – column 4 line 4); an identification of a new description file, the description file comprising code to be used to provide the new function (Nguyen et al., column 83 lines 38-48, and column 83 line 55 – column 84 line 5); and an identification of a new graphic symbol file for the new function, the graphic symbol file corresponding to a new graphic symbol to be displayed on the protocol tester to allow selection of the new function when setting up the communication procedure (Nguyen et al., column 35 lines 32-40, and column 36 line 66 – column 37 line 25); means for selecting instances involved in the communication procedure (Nguyen et al., column 60 lines 51-65); means for selecting a protocol layer based upon the selected communication procedure; means for selecting abstract communication interfaces of the protocol layer for the communication procedure (Nguyen et al., column 24 lines 22-45); means for selecting communication data; means for setting up the communication procedure through the protocol tester with the abstract communications interfaces and the communication data begin selected graphically (Nguyen et al., column 13 lines 18-32); means for selecting the new function while setting up the communication procedure by graphically selecting the new graphic symbol (Nguyen et al., column 8 lines 27-46 and column 8 lines 58-60); and means for associating parameters selectable for the

new function with the new description file (Nguyen et al., column 63 lines 10-36).

Consider claim 30, as applied to claim 29. Nguyen et al., as modified by Ehrhardt et al., discloses a protocol tester wherein for each parameter of the new function and for a result of the new function, entering into the configuration file a name and a data type (Nguyen et al., column 29 line 64 – column 30 line 2).

Consider claim 31, as applied to claim 29. Nguyen et al., as modified by Ehrhardt et al., discloses a protocol tester wherein the graphic symbol is selected from the group consisting of: a display name of the new function, and an icon for the new function, the graphic symbol file comprising display data for the graphic symbol (Nguyen et al., column 100 lines 38-64).

Consider claim 34, as applied to claim 29. Nguyen et al., as modified by Ehrhardt et al., discloses a protocol tester wherein data for the new functionality is entered in the configuration file (Nguyen et al., column 29 line 64 – column 30 line 2).

Consider claim 35, as applied to claim 29. Nguyen et al., as modified by Ehrhardt et al., discloses a protocol tester wherein the abstract communication interfaces comprise Service Access Points (SAP) (Nguyen et al., column 62 line 15-40).

7. Claims 20-21, 23-24, and 32-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nguyen et al. (US 6304915 B1) in view of Ehrhardt et al. (US 20010015732 A1) and in further view of Peart (US 2030074393 A1).

Consider claim 20, as applied to claim 17. Nguyen et al., as modified by Ehrhardt et al., discloses a system, method and article of manufacture for a gateway system architecture with system administration information accessible from a browser comprising a method of setting up a communication procedure that is executable on a protocol tester using message sequence charts (MSC) to graphically display the communication procedure between two instances, comprising: creating a configuration file to be used by the protocol tester, the configuration file including one or more new functions that are not available in an original version of the protocol tester provided by a manufacturer, the configuration file comprising for each of the new functions: a name for the new function; a data type for the new function; an identification of a new description file, the description file comprising code to be used to provide the new function; and an identification of a new graphic symbol file for the new function, the graphic symbol file corresponding to a new graphic symbol to be displayed on the protocol tester to allow selection of the new function when setting up the communication procedure; selecting instances involved in the communication procedure; selecting a protocol layer based upon the selected communication procedure; selecting abstract communication interfaces of the protocol layer for the communication procedure; selecting communication data; setting up the communication procedure through the protocol

tester based on the selections made in the selecting steps, with abstract communications interfaces selecting and the communication data selecting steps begin made graphically; selecting the new function while setting up the communication procedure by graphically selecting the new graphic symbol; and associating parameters selectable for the new function with the new description file. However, Nguyen et al., as modified by Ehrhardt et al., fails to disclose a method wherein the description file is formulated in Forth, Jscript or VBScript. Peart discloses methods for distributed program execution with file-type association in a client-server network comprising a method wherein the description file is formulated in Forth, Jscript or VBScript ("In another embodiment, the Program Neighborhood application allows scripts to access to information via an application programming interface. Scripts may be written in, for example, VBScript or JScript. In this embodiment, the scripting language is used to dynamically generate an output display using information returned by the application in response to queries posed by the script. Once the output display is generated, it is transmitted to client node 20 for display by the browser program 80. A user of the client node 20 can launch an application by selecting an icon 57, 57' displayed in the Program Neighborhood web page. In some embodiments, each icon 57, 57' is an encoded URL that specifies: the location of the application (i.e., on which servers it is hosted or, alternatively, the address of a master server); a launch command associated with the application; and a template identifying how the output of the application should be displayed (i.e., in a window "embedded" in the browser or in a separate window). In some embodiments, the URL includes a file, or a reference to a file, that contains the

information necessary for the client to create a connection to the server hosting the application. This file may be created by the Program Neighborhood application dynamically. The client node 20 establishes a connection (arrow 94) with the server identified as hosting the requested application (in this example, server 32) and exchanges information regarding execution of the desired application. In some embodiments, the connection 94 is made using the Independent Computing Architecture (ICA) protocol, manufactured by Citrix Systems, Inc. of Fort Lauderdale, Fla. Thus, the client node 20 may display application output in a window separate from the web browser 60, or it may "embed" application output within the web browser.") paragraphs 0073-0074).

Nguyen et al., as modified by Ehrhardt et al., discloses a prior art system, method and article of manufacture for a gateway system architecture with system administration information accessible from a browser comprising a method of setting up a communication procedure that is executable on a protocol tester using message sequence charts (MSC) to graphically display the communication procedure between two instances, comprising: creating a configuration file to be used by the protocol tester, the configuration file including one or more new functions that are not available in an original version of the protocol tester provided by a manufacturer, the configuration file comprising for each of the new functions: a name for the new function; a data type for the new function; an identification of a new description file, the description file comprising code to be used to provide the new function; and an identification of a new graphic symbol file for the new function, the graphic symbol file corresponding to a new

graphic symbol to be displayed on the protocol tester to allow selection of the new function when setting up the communication procedure; selecting instances involved in the communication procedure; selecting a protocol layer based upon the selected communication procedure; selecting abstract communication interfaces of the protocol layer for the communication procedure; selecting communication data; setting up the communication procedure through the protocol tester based on the selections made in the selecting steps, with abstract communications interfaces selecting and the communication data selecting steps begin made graphically; selecting the new function while setting up the communication procedure by graphically selecting the new graphic symbol; and associating parameters selectable for the new function with the new description file upon which the claimed invention can be seen as an improvement.

Pearl teaches a prior art comparable methods for distributed program execution with file-type association in a client-server network comprising a method wherein the description file is formulated in Forth, Jscript or VBScript.

Thus, the manner of enhancing a particular device (methods for distributed program execution with file-type association in a client-server network comprising a method wherein the description file is formulated in Forth, Jscript or VBScript) was made part of the ordinary capabilities of one skilled in the art based upon the teaching of such improvement in Pearl. Accordingly, one of ordinary skill in the art would have been capable of applying this known improvement technique in the same manner to the prior art system, method and article of manufacture for a gateway system architecture with system administration information accessible from a browser comprising a method

of setting up a communication procedure that is executable on a protocol tester using message sequence charts (MSC) to graphically display the communication procedure between two instances, comprising: creating a configuration file to be used by the protocol tester, the configuration file including one or more new functions that are not available in an original version of the protocol tester provided by a manufacturer, the configuration file comprising for each of the new functions: a name for the new function; a data type for the new function; an identification of a new description file, the description file comprising code to be used to provide the new function; and an identification of a new graphic symbol file for the new function, the graphic symbol file corresponding to a new graphic symbol to be displayed on the protocol tester to allow selection of the new function when setting up the communication procedure; selecting instances involved in the communication procedure; selecting a protocol layer based upon the selected communication procedure; selecting abstract communication interfaces of the protocol layer for the communication procedure; selecting communication data; setting up the communication procedure through the protocol tester based on the selections made in the selecting steps, with abstract communications interfaces selecting and the communication data selecting steps begin made graphically; selecting the new function while setting up the communication procedure by graphically selecting the new graphic symbol; and associating parameters selectable for the new function with the new description file of Nguyen et al., as modified by Ehrhardt et al., and the results would have been predictable to one of ordinary skill in

the art, namely, one skilled in the art would have readily recognized a method of linking communications among protocol layers.

Consider claim 21, as applied to claim 17. Nguyen et al., as modified by Ehrhardt et al. and Peart, discloses a method wherein the configuration file is implemented as a text file selected from the group consisting of an INI format and an XML format ((“FIG. 3D shows an exemplary process of communicating between the client node 20, a server 30 that in this example acts as a web server, and server 32. The client node 20 authenticates itself to the server 30 as described above in connection with FIG. 2C. In one embodiment, the server 30 accesses an output display template 90, such as an SGML, HTML or XML file, to use as a base for constructing the Program Neighborhood window to transmit to the client node 20. The template may be stored in volatile or persistent memory associated with the server 30 or it may be stored in mass memory 92, such as a disk drive or optical device, as shown in FIG. 3D. In this embodiment, the template 90 is a standard SGML, HTML, or XML document containing Program Neighborhood-specific tags that are replaced with dynamic information. The tags indicate to the server 30 where in the output display to insert information corresponding to available applications, such as icon images. In one particular embodiment, the Program Neighborhood-specific tags are embedded within comments inside the file, allowing the file to remain compatible with standard interpreters. In another embodiment, the Program Neighborhood-specific tags are extensions of the markup language used as the base for the template.”) Peart, paragraphs 0067-0068).

Consider claim 23, as applied to claim 12. Nguyen et al., as modified by Ehrhardt et al. and Peart, discloses a method wherein the configuration file further includes information on how many additional functions are stored in it, the additional functions not available in the original version of the protocol tester provided by a manufacturer (" [0074] A user of the client node 20 can launch an application by selecting an icon 57, 57' displayed in the Program Neighborhood web page. In some embodiments, each icon 57, 57' is an encoded URL that specifies: the location of the application (i.e., on which servers it is hosted or, alternatively, the address of a master server); a launch command associated with the application; and a template identifying how the output of the application should be displayed (i.e., in a window "embedded" in the browser or in a separate window). In some embodiments, the URL includes a file, or a reference to a file, that contains the information necessary for the client to create a connection to the server hosting the application. This file may be created by the Program Neighborhood application dynamically. The client node 20 establishes a connection (arrow 94) with the server identified as hosting the requested application (in this example, server 32) and exchanges information regarding execution of the desired application. In some embodiments, the connection 94 is made using the Independent Computing Architecture (ICA) protocol, manufactured by Citrix Systems, Inc. of Fort Lauderdale, Fla. Thus, the client node 20 may display application output in a window separate from the web browser 60, or it may "embed" application output within the web browser.") Peart, paragraph 0074).

Consider claim 24, as applied to claim 17. Nguyen et al., as modified by Ehrhardt et al. and Peart, discloses a method wherein a reading-in of the description file occurs via an include command (`(#include"rr.h")`) Nguyen et al., column 66 lines 10-15 ("A user of the client node 20 can launch an application by selecting an icon 57, 57' displayed in the Program Neighborhood web page. In some embodiments, each icon 57, 57' is an encoded URL that specifies: the location of the application (i.e., on which servers it is hosted or, alternatively, the address of a master server); a launch command associated with the application; and a template identifying how the output of the application should be displayed (i.e., in a window "embedded" in the browser or in a separate window). In some embodiments, the URL includes a file, or a reference to a file, that contains the information necessary for the client to create a connection to the server hosting the application. This file may be created by the Program Neighborhood application dynamically. The client node 20 establishes a connection (arrow 94) with the server identified as hosting the requested application (in this example, server 32) and exchanges information regarding execution of the desired application. In some embodiments, the connection 94 is made using the Independent Computing Architecture (ICA) protocol, manufactured by Citrix Systems, Inc. of Fort Lauderdale, Fla. Thus, the client node 20 may display application output in a window separate from the web browser 60, or it may "embed" application output within the web browser.") Peart, paragraph 0074 ("Still referring to FIG. 9A, the client node presents a user with a graphical depiction of at least one data file stored on at least one server node (Step

200). The graphical depiction of the data files includes but is not limited to the display of graphical indicia such as icons or the display of file names. In one embodiment, indicia representative of files stored on one or more server nodes are intermingled with indicia representative of files stored on the client node. In this embodiment, server-based FTA is operative when indicia representative of a file stored on a server node is selected. In another embodiment, multiple forms of FTA (see above, below) are operative, with the appropriate form of FTA activated based on the location of the file associated with the selected graphical indicia.") Peart, paragraphs 0074-0149).

Consider claim 32, as applied to claim 29. Nguyen et al., as modified by Ehrhardt et al. and Peart, discloses a protocol tester wherein the description file is formulated in Forth, Jscript or VBScript (Peart, paragraph 0149).

Consider claim 33, as applied to claim 29. Nguyen et al., as modified by Ehrhardt et al. and Peart, discloses a protocol tester wherein the configuration file is implemented as a text file selected from the group consisting of an INI format (Nguyen et al., source code, column 67)and an XML format (Peart, paragraphs 0067-0068).

8. Claims 27 and 36 are is rejected under 35 U.S.C. 103(a) as being unpatentable over Nguyen et al. (US 6304915 B1) in view of Ehrhardt et al. (US 20010015732 A1) and in further view of Hayduk et al. (US 20030112790 A1).

Consider claim 27, as applied to claim 17. Nguyen et al., as modified by Ehrhardt et al., discloses a system, method and article of manufacture for a gateway system architecture with system administration information accessible from a browser comprising a method of setting up a communication procedure that is executable on a protocol tester using message sequence charts (MSC) to graphically display the communication procedure between two instances, comprising: creating a configuration file to be used by the protocol tester, the configuration file including one or more new functions that are not available in an original version of the protocol tester provided by a manufacturer, the configuration file comprising for each of the new functions: a name for the new function; a data type for the new function; an identification of a new description file, the description file comprising code to be used to provide the new function; and an identification of a new graphic symbol file for the new function, the graphic symbol file corresponding to a new graphic symbol to be displayed on the protocol tester to allow selection of the new function when setting up the communication procedure; selecting instances involved in the communication procedure; selecting a protocol layer based upon the selected communication procedure; selecting abstract communication interfaces of the protocol layer for the communication procedure; selecting communication data; setting up the communication procedure through the protocol tester based on the selections made in the selecting steps, with abstract communications interfaces selecting and the communication data selecting steps begin made graphically; selecting the new function while setting up the communication procedure by graphically selecting the new graphic symbol; and associating parameters

selectable for the new function with the new description file. However, Nguyen et al., as modified by Ehrhardt et al., fails to disclose a method wherein the communication data is selected from the group consisting of Protocol Data Units (PDU) and Abstract Service Primitives (ASP). Hayduk et al. discloses providing data link services to upper layer protocols in application and communication subsystems wherein communication data is selected from the group consisting of Protocol Data Units (PDU) ("The unacknowledged mode data transfer connection management entities (UM CMES) 22b are automatically assigned a CEI 25a during the channel setup. The CEI 25a is returned when the channel is released. The UM CMEs 22b implement connection oriented data transmission through a logical data link connection or higher layer protocol data units (PDUs).") paragraph 0025) and Abstract Service Primitives (ASP) ("In accordance with some embodiments of the present invention, functionality can move across processors or physical communication interfaces may change without affecting the software for the data link layers. This is due to the fact that the PLC 26 handles this dependency, allowing the LLC 20 and the higher layers to move freely. The abstract layer of the service primitive hides these physical dependencies.") paragraph 0036).

Nguyen et al., as modified by Ehrhardt et al., discloses a prior art system, method and article of manufacture for a gateway system architecture with system administration information accessible from a browser comprising a method of setting up a communication procedure that is executable on a protocol tester using message sequence charts (MSC) to graphically display the communication procedure between two instances, comprising: creating a configuration file to be used by the protocol tester,

the configuration file including one or more new functions that are not available in an original version of the protocol tester provided by a manufacturer, the configuration file comprising for each of the new functions: a name for the new function; a data type for the new function; an identification of a new description file, the description file comprising code to be used to provide the new function; and an identification of a new graphic symbol file for the new function, the graphic symbol file corresponding to a new graphic symbol to be displayed on the protocol tester to allow selection of the new function when setting up the communication procedure; selecting instances involved in the communication procedure; selecting a protocol layer based upon the selected communication procedure; selecting abstract communication interfaces of the protocol layer for the communication procedure; selecting communication data; setting up the communication procedure through the protocol tester based on the selections made in the selecting steps, with abstract communications interfaces selecting and the communication data selecting steps begin made graphically; selecting the new function while setting up the communication procedure by graphically selecting the new graphic symbol; and associating parameters selectable for the new function with the new description file upon which the claimed invention can be seen as an improvement.

Hayduk et al. teaches a prior art method of providing data link services to upper layer protocols in application and communication subsystems wherein communication data is selected from the group consisting of Protocol Data Units and Abstract Service Primitives.

Thus, the manner of enhancing a particular device (method of providing data link services to upper layer protocols in application and communication subsystems wherein communication data is selected from the group consisting of Protocol Data Units and Abstract Service Primitives) was made part of the ordinary capabilities of one skilled in the art based upon the teaching of such improvement in Hayduk et al. Accordingly, one of ordinary skill in the art would have been capable of applying this known improvement technique in the same manner to the prior art system, method and article of manufacture for a gateway system architecture with system administration information accessible from a browser comprising a method of setting up a communication procedure that is executable on a protocol tester using message sequence charts (MSC) to graphically display the communication procedure between two instances, comprising: creating a configuration file to be used by the protocol tester, the configuration file including one or more new functions that are not available in an original version of the protocol tester provided by a manufacturer, the configuration file comprising for each of the new functions: a name for the new function; a data type for the new function; an identification of a new description file, the description file comprising code to be used to provide the new function; and an identification of a new graphic symbol file for the new function, the graphic symbol file corresponding to a new graphic symbol to be displayed on the protocol tester to allow selection of the new function when setting up the communication procedure; selecting instances involved in the communication procedure; selecting a protocol layer based upon the selected communication procedure; selecting abstract communication interfaces of the protocol

layer for the communication procedure; selecting communication data; setting up the communication procedure through the protocol tester based on the selections made in the selecting steps, with abstract communications interfaces selecting and the communication data selecting steps begin made graphically; selecting the new function while setting up the communication procedure by graphically selecting the new graphic symbol; and associating parameters selectable for the new function with the new description file of Nguyen et al., as modified by Ehrhardt et al., and the results would have been predictable to one of ordinary skill in the art, namely, one skilled in the art would have readily recognized a method of linking communications among protocol layers.

Consider claim 36, as applied to claim 29. Nguyen et al., as modified by Ehrhardt et al. and Hayduk et al., discloses a protocol tester wherein the communication data is selected from the group consisting of Protocol Data Units (PDU) (Hayduk et al., paragraph 0025) and Abstract Service Primitives (ASP) (Hayduk et al., paragraph 0036).

Response to Arguments

9. Applicant's arguments filed 23 March 2009 with respect to claims 17 and 29 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

10. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

Any response to this Office Action should be faxed to (571) 273-8300 or mailed to:

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Hand-delivered responses should be brought to

Customer Service Window

Randolph Building
401 Dulany Street
Alexandria, VA 22314

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Mark Fearer whose telephone number is (571) 270-1770. The Examiner can normally be reached on Monday-Thursday from 7:30am to 5:00pm.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tonia Dollinger can be reached on (571) 272-4170. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free) or 571-272-4100.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist/customer service whose telephone number is (571) 272-2600.

Mark Fearer
/M.D.F./
June 30, 2009

/George C Neurauter, Jr./

Application/Control Number: 10/692,770

Page 36

Art Unit: 2443

Primary Examiner, Art Unit 2443